The JavaScript Array

*Use this* [*link*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Indexed_collections#Array_object) *as the main reference*

The following questions are meant to introduce the JavaScript array, the only built-in list type in JavaScript (pre ES6), and the functionality it offers.

Whenever you encounter a new language the very first thing you usually should focus on is its collection framework.

All questions are one-liners, so the main task is to figure out which method to use (hints are given for each question)

**a)** Create the two arrays below, spelled exactly as they are given. This will form the start for all the following questions.

var boys = ["Peter", "lars", "Ole"];

var girls = ["Janne", "hanne", "Sanne"];

*The array type has a method* [*concat()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/concat) *to merge two or more arrays (does* ***concat*** *mutate an existing array?)*

**b)** Create a new array called ***all***, which should be a concatenation of the two arrays given above, starting with the boys and ending with the girls.

**c)** *The array type has a cool method to reduce an array into a single string* [*join(*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/join)*) (you will love this method)*

· Create a comma separated string containing all the names from the *all*-array, separated by commas.

· Create a hyphen (-) separated string containing all the names from the *all*-array, separated by hyphens.

*The array type provides methods to* ***add*** *items to the start -* [*unshift()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/unshift) *and to the end* [*push(..)*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push) *of an array.*

**d)** Add the names Lone and Gitte to the end of the array (remember, all can be done in one-liners)

**e)** Add the names Hans and Kurt to the start of the array

*The array type provides methods to* ***remove*** *items from the start* [*shift()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/shift) *and from the end* [*pop(..)*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/pop) *of an array.*

**f)** Remove the first name in the array (Hans)

**g)** Remove the last name from the array (Gitte)

*The array type provides a method* [*splice(..)*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice) *which changes the array by removing existing elements and/or adding new*

**h)** Remove Ole and Janne from the middle of the array

*The array type provides a method* [*reverse()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reverse) *to reverse the elements of an array*

**i)** Sanne thinks it’s unfair that the boys have to come first, reverse the *all* array, so that the girls come first.

*The array type provides a method* [*sort()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort) *to sort the elements of an array*

**j)** Peter thinks that this is just as unfair and suggests that the array should be sorted. Sort the array.

**k)** The default sort algorithm doesn’t handle the situation where the name can be either capitalized or not. Write a user-defined sort method to fix this problem.

**Array methods with callbacks (there are many)**

*The array type provides a method* [*map()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map) *which* returns a new array of the return value from executing the callback on every array item**.**

**l)** Convert all the names in the array to uppercase.

*The array type has a method* [*filter()*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter) *that creates a new array with all elements that pass the test implemented by the provided callback*

**m**) Create a new array containing all the names that start with either “l” or “L” (hint: use the filter function with a sufficient callback).